

Amendments to the Claims:

1-8. **(canceled)**

9. **(previously presented)** A system for testing an application running on a target device, the system comprising:

a target device storing and executing a software test agent, wherein said application under test is also stored and executed on said target device;

a test development computer storing and executing a software test tool for testing and validating said application's rendering of output to a Graphical User Interface (GUI), said test tool configured to communicate with said agent on said target device;

said software test tool executing a test script for testing and validating one or more aspects of said application's rendering of output to said GUI, said test tool operable to generate requests to said agent to obtain information from and send events to controls associated with said GUI;

said software test tool comprising a configuration manager for handling testing of said application against multiple languages and platform configurations;

said configuration manager comprising a configuration table having a plurality of user-defined configurations, each said configuration comprising a collection of value sets corresponding to respective configuration item groups, each said value set being one of a plurality of possible value sets that are selectable in association with said respective item group, each said value set comprising a collection of one or more related configuration items and corresponding values for said related items; and

said configuration table being stored in a spread sheet format for ease of editing by a user, said spread sheet being configured for allowing said user to simultaneously view in a tabular format said plurality of possible value sets that are selectable in association with each said configuration item group wherein,

in response to a user changing from a first value set to a second value set during execution of said test script, said software test tool is further configured to:

automatically delete all configuration item values associated with said first value set and reload said configuration items with corresponding values associated with said second value set; and

continue executing said test script using said configuration item values associated with said second value set.

10-12. **(canceled)**

13. **(previously presented)** The system of claim 9, wherein said configuration manager comprises a capability that allows said user to get said value of a particular configuration item, and a capability that allows said user to set said value of a particular configuration item, during execution of said test script.

14. **(previously presented)** The system of claim 9, wherein said test tool comprises a capability that allows said user to assign an input-event name to a grouping of multiple key sequences, said grouping of multiple key sequences representing an input event that occurs on said target device during execution of said test script, wherein said input-event name can be used in place of said grouping each time said grouping is to be written into said test script.

15. **(previously presented)** A method of testing an application that is operable to execute in multiple languages and platform configurations, the method comprising:

storing a configuration table in a spread sheet format for ease of editing by a user having a plurality of user-defined configurations, each configuration comprising a collection of value sets corresponding to respective configuration item groups, each said value set being one of a plurality of possible value sets that are selectable in association with said respective item group, each said value set comprising a collection of one or more related configuration items and corresponding values for said related items, and said spread sheet being configured for allowing said user to simultaneously view in a tabular format said plurality of possible value sets that are selectable in association with each said configuration item group;

writing a test script that executes differently based on which user-defined configuration is loaded from said table;

loading a user-defined configuration from said configuration table prior to execution of said test script; and

executing said test script in accordance with said configuration loaded from said configuration table wherein,

in response to changing from a first value set to a second value set during execution of said test script:

automatically deleting all configuration item values associated with said first value set;

automatically reloading said configuration items with corresponding values associated with said second value set; and

continue executing said test script using said configuration item values associated with said second value set.

16-18. **(canceled)**

19. **(previously presented)** A method performed by a test development computer storing and executing a test tool operable to communicate with a target device storing and executing a test agent to interrogate a particular Graphical User Interface (GUI) control that is associated with an application under test running on said target device, the method comprising:

storing a first table that comprises a list of entries accessible by said test tool, each said entry containing information corresponding to a particular GUI control associated with said application under test, said information associated with said particular GUI control comprising:

a user-defined name for uniquely referring to said control;

a plurality of properties which define said control, including a class name that indicates a class type to which said control belongs; and

a data field for optionally storing a control-specific identify flag to be used by said test tool for indicating to said test agent that a specific set of properties are to be used in identifying said control on said target device; and

storing a second table that comprises a list of entries accessible by said test tool, each said entry containing information related to a particular class of GUI controls associated with said application under test, said information associated with each said particular class of GUI controls comprising:

a class name for uniquely referring to said particular class of controls; and

a default identify flag to be used by said test tool for indicating to said test agent that a default set of properties are to be used for identifying controls of said particular class of controls on said target device that do not have a control-specific identify flag associated with them, as defined in said first table.

20. **(previously presented)** The method of claim 19, further comprising:

- locating in said first table said user defined name of said control to be interrogated;
- reading from said first table said properties associated with said control to be interrogated;
- checking to see whether said control-specific identify-flag data field associated with said control is empty;
- loading said identify flag from said control-specific identify flag data field, if said data field is not empty;
- if said control-specific identify-flag field is empty, loading said default identify flag from said second table using said class name associated with said control to be interrogated;
- generating a request to said agent to locate a control on said target device that matches said control to be interrogated, said agent locating a matching control by searching for a control which has a set of properties indicated by said identify flag obtained in said steps above that match said corresponding set of properties for said control to be interrogated, said matching control having a handle for accessing said control associated therewith; and
- receiving said handle for said matching control from said agent.

21. **(currently amended)** The method of claim 20, wherein said entry associated with each said control listed in said first table further comprises a data field for optionally storing a control-specific verify flag, said control-specific verify flag used for indicating that a specific set of properties are to be used when verifying that an occurrence of said control on said target device matches an expected state; and

wherein said entry associated with each said class of controls listed in said second table further comprises a default verify flag, said default verify flag used for indicating that a default set of properties are to be used for verifying controls of said particular class of controls on said target device that do not have a control-specific verify flag associated with them, as defined in said first table.

22. **(currently amended)** The method of claim 21, further comprising:

checking to see whether said control-specific verify-flag data field associated with said control to be interrogated is empty;

loading said verify flag from said control-specific verify flag data field, if said data field is not empty;

if said control-specific verify-flag field is empty, loading said default verify flag from said second table using said class name associated with said control to be interrogated;

interpreting said verify flag obtained in said steps above to determine which properties are to be used in verifying that said matching control matches said control to be interrogated;

generating a request to said agent to retrieve actual property values associated with said matching control, said values retrieved comprising values for at least those properties indicated in said verify flag obtained in said steps above; and

comparing said actual property values associated with said matching control, for only those properties indicated in said verify flag obtained above, to **said** [[v]] corresponding property values defined in said first table for said control to be interrogated.

23. **(previously presented)** The method of claim 22, further comprising:

upon detecting a mismatch in any of said properties compared, generating an entry in a log file detailing said mismatch.